

UNITED STATES PATENT APPLICATION
FOR

POINT-OF-PRESENCE CALL CENTER
MANAGEMENT SYSTEM

INVENTORS:

Prem Uppaluru
Mukesh Sundaram

Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CALIFORNIA 90025
(408) 720-8598

Attorney Docket No. 003924.P001DC

"Express Mail" mailing label number: EL672751456US

Date of Deposit: January 15, 2002

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

P. Baloro

(Typed or printed name of person mailing paper or fee)

P. Baloro
(Signature of person mailing paper or fee)

1/15/02
(Date signed)

POINT-OF-PRESENCE CALL CENTER MANAGEMENT SYSTEM

[0001] This is a continuation of co-pending U.S. Patent application no. 09/433,640, filed on November 3, 1999, and entitled, "Point-of-Presence Call Center Management System."

FIELD OF THE INVENTION

[0002] The present invention relates to the field of telecommunication, and more particularly to management of toll free telephone calls.

BACKGROUND OF THE INVENTION

[0003] Fig. 1 is a functional diagram of a premises call center connecting an end user 116 to a business call center 108 via an originating Local Public Switched Telecommunications Network (PSTN) 106, a Long Distance Network 114 and terminating Local PSTN 106. Business call centers are typically put together by integrating multiple system components into a complete business solution to answer, service, queue and route inbound customer calls. These system components can include a Private Branch Exchange (PBX) 102, an Automatic Call Distributor (ACD) 112 and an Interactive Voice Response (IVR) System 110 in addition to customer service or help desk applications for the call center agents 104. Many call centers deploy a Computer Telephony Integration (CTI) server providing intelligent call routing. Traditionally, different vendors supplied the different system components and systems integrators pulled the components together into a solution.

[0004] Fig. 2 is a functional diagram of a network-based call center connecting an end user 116 to a business call center 108 via an originating Local PSTN 106, a Long Distance Network 114 and a terminating Local PSTN 106. Network call centers may

include a Switch 122, an ACD 112 and an IVR 110 within the Long Distance Network 114 and provide call answering, servicing and queuing services. These services are built on call center solutions residing inside the network that aggregate the services across multiple business customers on the shared physical configurations. Many call center vendors have targeted this fast growing network call center market with PSTN integrated systems and solutions.

[0005] The call centers depicted in Figs. 1 and 2 each share the disadvantage that long distance toll charges accrue while a call is on hold awaiting connection to a call center agent. Long distance toll charges also accrue while the caller is interacting with the Interactive Voice Response.

SUMMARY OF THE INVENTION

[0006] A method and system for managing a toll free long distance call to a business call center are disclosed. A toll free long distance call to a business call center is redirected to a local call center. The redirected toll free call is automatically answered in the local call center to determine whether long distance connection to the business call center is necessary. If connection to the remote call center is necessary, the redirected toll free call is bridged with a telephone connection in the business call center via a long distance network.

BRIEF DESCRIPTION OF THE DRAWING

[0007] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawing in which like references indicate similar elements and in which:

[0008] Fig. 1 is a schematic diagram of a prior art call center configuration with PBX, ACD and IVR systems located at the business call center;

[0009] Fig. 2 is a schematic diagram of a prior art network based call center configuration with Switch, ACD and IVR systems located inside the long distance network;

[0010] Fig. 3 is a schematic diagram of a POP Call Center configuration according to an embodiment that includes Switch and POP Call Center Gateway located at points of presence connected to Premises Call Center Gateway over a Call Center Network;

[0011] Fig. 4 is a schematic diagram of components of a POP Call Center system according to one embodiment, including a POP Call Center Gateway, a Premises Call Center Gateway and a Call Center Network of a business;

[0012] Fig. 5 is a schematic diagram of components of a POP Call Center system according to an embodiment that supports a single business with multiple call center sites connected with multiple POP Call Centers;

[0013] Fig. 6 is a schematic diagram of components of a POP Call Center system according to an embodiment that supports multiple business call centers connected to multiple POP Call Centers; and

[0014] Fig. 7 is a schematic diagram of POP Call Center System software modules and their interconnections according to one embodiment.

DETAILED DESCRIPTION

[0015] Although the present invention is described below by way of various embodiments that include specific structures and methods, embodiments that include alternative structures and methods may be employed without departing from the principles of the invention described herein.

OVERVIEW OF EMBODIMENTS

[0016] In general, embodiments described below feature a global call center system capable of answering, servicing, queuing and routing of calls at local points of presence to reduce communications costs and enhance operational efficiency for toll-free inbound call centers. In at least one embodiment, the global call center system includes a set of point-of-presence call center gateways distributed at points of presence close to the point of call origination that are interconnected by a virtual private network to premises call center gateways at business locations where the call centers reside.

[0017] A point-of-presence (POP) call center gateway according to embodiments of the invention is capable of intercepting and answering inbound toll-free telecommunications calls at or near the point of call origination. The POP call center gateway is also capable of providing interactive voice response based automated service, holding and queuing the calls until operators are available to service the call, and playing music or customized announcements to the caller while the call is being held. The POP call center gateway is further capable of requesting connected premises call center gateways to originate proxy calls on its behalf, monitoring call progress and routing the locally queued calls to the premises call center just in time before the operator picks up the call.

[0018] A premises call center gateway according to embodiments of the invention is capable of receiving proxy call origination requests from connected POP call center gateways and in response, generating and presenting proxy calls to the automatic call distributor (ACD) at the premises call center. The premises call center gateway is further capable of monitoring the progress of such proxy calls within the ACD for operator availability, communicating with the appropriate POP call center gateways for just in time call delivery to the selected operator, and bridging the calls between the POP call center gateways and the premises ACD.

[0019] Embodiments of the invention include a system and method for operating the global call center system where in a customer calls an advertised inbound toll-free number to reach a business call center. The call is intercepted at the local central office or tandem switch and routed to a local POP call center gateway using the point of call routing capability of a Service Management System/800 (SMS/800) database system and signaling system 7 (SS7) protocols. The local central office or tandem switch uses the SMS/800 database system to translate the single inbound toll-free number for the business call center into a matching local phone number terminating at the local POP call center gateway. The local central office or tandem switch identifies to the local POP call center gateway the translated called party number and optionally the calling party number. The call is terminated at the local POP call center gateway, which answers the call with an automated voice response system.

[0020] The POP call center gateway uses the translated called party number to identify and run a matching interactive voice response application customized to the business call center. The customized interactive voice response application can either be replicated at each local POP call center gateway or dynamically accessed from the

business call center just in time as the call is answered and serviced. In either case, the POP call center gateway responds to the incoming call with an interactive voice response application customized to the business call center that was called by the customer. This custom interactive voice response application then services the customer call by providing appropriate prompts and menus, gathering input from the customer and interactively providing automated service. The custom interactive voice response application uses a virtual private network (connecting the POP call center gateways to one or more business premises call center gateways) to securely access the customer application and/or data at the corresponding business call center to appropriately service the calling customer.

[0021] If the call is to be held waiting for an available call center agent, the POP call center gateway holds and queues the call locally while requesting the corresponding premises call center gateway to insert a proxy call into the business call center's ACD. The POP call center gateway, optionally plays music and/or periodic prompts and messages to the caller while the call is on hold. The corresponding premises call center gateway inserts a proxy call in the business call center's ACD and starts monitoring its progress through the ACD queue.

[0022] When the proxy call reaches the head of the queue and is about to be answered by a live call center agent, the premises call center gateway alerts the waiting POP call center gateway. The waiting POP call center gateway then routes the locally queued call to the premises call center gateway over an appropriate long distance network, either a public/private switched telephone network or a public/private Internet Protocol (IP) telephony network. The corresponding premises call center gateway receives the routed

call, matches it with the corresponding proxy call to the business call center ACD and bridges the incoming call to the proxy call.

[0023] A selected call center agent at the business call center then answers the call and provides expected customer service to the customer. Finally, when the customer or the call center agent hangs up the call, the appropriate call center gateway detects the event and alerts the matching counter-part gateway and both gateways terminate the call.

SYSTEM DESCRIPTION

[0024] Fig. 3 is a functional diagram of a point-of-presence (POP) call center system in accordance with at least one embodiment of the present invention wherein the end user 116 is connected to the POP-enabled business call center 150 via an originating Local PSTN 106, a Long Distance Network 114 and a terminating Local PSTN 106.

[0025] The POP call center system extends the conventional premises and network based call center systems to a fully distributed call center system with multiple points of presence. The POP call center system is capable of locally answering, servicing, queuing and routing inbound toll-free calls to business call centers thereby saving on communications costs and increasing operating efficiency.

[0026] The POP call center system consists of one or more POP call center gateway servers 146 distributed at one or more points of presence 152 close to the points of call origination. These POP call center gateway servers 146 are connected by one or more call center networks 148 to premises call center gateway servers 142 at one or more POP-enabled business call centers 150. The POP call center gateway server 146 is connected to a Switch 144 enabling it to receive and originate calls on the local PSTN 106. The POP call center gateway servers 146 are further connected to a switched or dedicated

access public telecommunications network 114 enabling long distance voice communications with connected premises call center gateway servers.

[0027] A POP-enabled business call center 150 consists of one or more premises call center gateway servers 142, one of which would be selected dynamically at the time of handling of an incoming call at a POP call center gateway.

[0028] Referring to Fig. 4, a POP call center gateway 166 intercepts and answers inbound toll-free calls at or near their point of call origination. In addition, it provides automated service with interactive voice response applications, holds and queues the calls until appropriate operators are available to service the call, and plays music or customized announcements to the caller while the call is on hold. If a call is queued, this gateway further requests a corresponding premises call center gateway 164 to originate a proxy call at the call center ACD on its behalf and monitor the progress of the queued call. When the premises call center gateway 164 alerts the POP call center gateway 166, the POP call center gateway 166 routes the locally queued call to the premises call center 150 just in time before the operator picks up the call.

[0029] The premises call center gateway 164 responds to requests for call center information and applications from POP call center gateways 166, accesses the requested information and applications from premises call center database systems and supplies it to the requesting POP call center gateway 166. The premises call center gateway 164 further receives proxy call origination requests from the POP call center gateways 166 and generates proxy calls on their behalf to the premises call center automatic call distributor (ACD). The premises call center gateway 164 then monitors the progress of proxy calls within the ACD for operator availability, communicates with the appropriate originating POP call center gateway 166 for just in time call delivery to the selected

operator, and bridges the calls between the POP call center gateway 166 and the premises ACD.

[0030] Referring to Fig. 5, a call center network according to one embodiment is a virtual private network connecting the POP call center gateways to one or more premises call center gateways all of which belong to a single business call center. A virtual private network offers industry standard connection and transport protocols such as ATM, Frame Relay or Internet Protocol (IP) for secure and private data communications between connecting entities with optional quality of service guarantees. Referring to Fig. 6, each POP call center gateway can be part of multiple such call center networks one for each business call center that it serves. POP call center gateways use a call center network to connect to corresponding premises call center gateways and access appropriate interactive voice response applications and information as well as request proxy call origination and monitoring of call progress. A call center network can optionally support voice communications over ATM, Frame Relay or IP protocols. In such a case, the POP call center gateways can use the call center network as an alternative long distance voice communications network when calls are bridged across the premises call center gateway to the business call center ACD.

[0031] Referring to Fig. 7, all the call center networks connect to a global POP call center network directory service 194 for translating the called party number of an incoming call at a POP call center gateway to the network address of a corresponding premises call center gateway. For each called party number at each POP call center, the POP call center network directory maintains a service record containing at a minimum the corresponding premises call center gateway network address. In one preferred

embodiment of the invention, a POP call center network directory service uses a network directory based on the Internet standard Lightweight Directory Access Protocol (LDAP).

[0032] A POP call center gateway further comprises POP call manager 182, POP voice response client 184 and POP network manager 186 software modules hosted on an industry-standard computer telephony server. A computer telephony server consists of an industry standard server computer such as an Intel PC server or Sun Microsystems server enhanced with telephony and voice processing capabilities and running an industry standard applications server operating system such as Microsoft Windows NT or Sun Microsystems Solaris. In an alternative preferred embodiment, a POP call center gateway can comprise an IP telephony gateway server and a separate applications server connected over a high-speed local area network. An IP telephony gateway is capable of translating traditional circuit switched voice communications to packet switched communications and transporting voice over long distance using IP networks. In such a configuration, the applications server hosts the POP call manager, POP voice response client and POP network manager modules which interact with the IP telephony gateway for voice communications and signaling.

[0033] A premises call center gateway further comprises premises call manager 188, premises voice response server 190 and premises network manager 186 software module hosted on an industry standard computer telephony server similar to the one hosting the POP call center gateway. In an alternative preferred embodiment, a premises call center gateway can comprise an IP telephony gateway server and a separate applications server connected over a high-speed local area network. In such a configuration, the applications server hosts the premises call manager, premises voice response server and premises

network manager software modules which interact with the IP telephony gateway for voice communications and signaling.

[0034] For each participating business call center network, the POP call center system assigns a unique universally accessible inbound toll-free number. This number can be a previously existing 800/888 toll-free access number of a participating business call center. Depending on the geographic areas in which it wishes to receive POP call center service, the participating business call center chooses one or more POP call centers to be connected to its call center network. The POP call center system then assigns a distinct direct inward dial (DID) number for each POP call center connected to the business call center network. This DID number, also referred to as the POP call center called party number, uniquely identifies at each POP call center the specific business call center to which an incoming call is targeted. The POP call center gateway uses this called party number to identify the network address of the corresponding premises call center gateway.

[0035] The POP call center system uses the point of call routing capability of the SMS/800 database management system to route toll-free inbound calls originating in a local PSTN to the nearest and most cost effective POP call center capable of handling these calls. Local exchange carriers use SMS/800 database management system to intelligently route inbound toll-free calls to appropriate inter exchange carriers or other competitive local exchange carriers. For each inbound toll-free call targeted at a toll-free number, the responsible central office switch or tandem switch requests routing instructions from the SMS/800 database management system utilizing Signaling System 7 (SS7) protocols. The local PSTN switch then routes the call to the appropriate carrier based on the response received from the SMS/800 database management system. The

POP call center system programs the SMS/800 database management system to instruct the local PSTN switch to route the call to the appropriate POP call center using its uniquely assigned DID number matching the originally called toll-free number.

[0036] A POP call center gateway receives and terminates calls originating from a connected local public switched telecommunications network (PSTN) enabling it to locally answer, service and queue the calls. A POP call center gateway can be connected to local PSTN at a central office switch, a tandem switch or a LATA tandem switch depending on local telecommunications traffic patterns and geographic location of the PSTN switches. It should be noted that the type and location of PSTN switch to which POP call center gateway is connected determines the local communication costs and geographic coverage for the inbound toll-free calls. In general, connection at a higher level of the PSTN switch results in broader geographic coverage and higher local communications costs.

EXEMPLARY PSEUDO CODE LISTINGS

[0037] Pseudo code listings A-D are appended to and form part of this specification. The listings present pseudo code representations of the interactions between the POP modules and the premises modules, particularly the call manager and voice response components. The functionality of each module is described as a set of messages received by the module from other modules and the actions taken by the module in response to these messages. Each module maintains state through private data structures that are identified in the pseudo code.

POP Call Manager

[0038] Referring to pseudo code listing A, appended hereto, a POP call manager receives an incoming toll-free call through the message INCOMING_CALL and requests

the local POP network manager to identify the called party number and locate the business call center to which the call is directed by calling TranslateNumberToAddress. The POP network manager implements the translation. The POP call manager determines whether the corresponding call center is able to receive additional incoming calls by sending the message, ALLOCATE_PROXY_CALL, to the premises call manager. If the corresponding call center is able to receive additional incoming calls, the POP call manager attaches the incoming call on an available voice port and transfers the call to the POP voice response client by calling CreateVoiceResponseClientInstance with the operation parameter, ANSWER. If the corresponding call center is unable to receive further calls, the POP call manager generates a busy signal to the local PSTN. If the POP voice response client transfers the call back to it for queuing through the message QUEUE_CALL, the POP call manager requests the corresponding premises call center gateway to originate a proxy call at the premises ACD on its behalf by sending the message PLACE_PROXY_CALL. Upon completion, the POP call manager places the call on hold by transferring the call to the POP voice response client by calling CreateVoiceResponseClientInstance with the operation parameter, HOLD. When the premises call manager alerts the POP call manager that the call is about to be answered by an operator through the message AGENT_READY, the POP call manager terminates the voice response client by sending it the message, TERMINATE.. This results in the voice response client passing control of the call back to the POP call manager through the message TRANSFER_CALL. The POP call manager then routes the call over an appropriate long distance voice communications network to the premises call center gateway by calling PlaceCall, bridging the inbound call with the newly placed call by calling BridgeCall. At any time during the call, if the POP call manager receives a

termination message USER_TERMINATION from the POP voice response client, which sends such a message if the user terminates the call, it notifies the premises call manager of the event by sending the message TERMINATE_CALL. The POP call manager may also receive a notification from the premises call manager that the agent has terminated the call, through the message AGENT_TERMINATION. In either case, it performs clean up of the incoming and long distance voice ports, and all state data associated with the incoming call.

POP Voice Response Client

[0039] Referring to pseudo code listing B, appended hereto, a POP voice response client receives and responds to instructions from the local POP call manager to answer and service an incoming call to a specified business call center. Based on the instructions as well as the configuration options for the specified business call center, the POP voice response client locates and connects to the premises voice response server on the matching premises call center gateway. Thereafter, the POP voice response client interacts with the corresponding premises voice response server to run an interactive voice response application customized to the business call center. Such interactions include accessing the necessary voice prompts, menus, forms, scripts, data and applications from the premises voice response server. It should be noted that the interactive voice response application can be customized to the specified business center by previously loading all the necessary voice prompts, menus, forms, scripts, and applications at the POP call center gateway. This approach would require full replication of all business applications at all POP call centers, which is wasteful in utilization of resources and expensive due to operational complexity. Thus, while business applications may be replicated at all POP call centers in certain embodiments, the

envisioned approach is to use a distributed voice user interface manager that adapts dynamically to the required interactive application customized to the specified business call center.

[0040] A preferred embodiment of the invention features a POP voice user interface manager embedded in the POP voice response client enabling it to dynamically adapt to the specified business call center's interactive voice response application without having to locally store all business call center applications at each POP call center. A POP voice user interface manager dynamically accesses voice prompts, menus, forms, scripts and applications customized to a specified business call center as needed from the corresponding premises voice response server. The POP voice user interface manager and the corresponding premises voice response server use a specialized request/response protocol such as the Internet standard Hyper Text Transfer Protocol to access the distributed resources. Optionally, the POP voice user interface manager and the corresponding premises voice response server may use the Internet standard Hyper Text Markup Language (HTML) or its extensions such as Extended Markup Language (XML) to access conveniently packaged units of information or application across the call center virtual private network. The POP voice response client optionally stores frequently used and rarely modified voice prompts and messages locally in a network cache to improve access efficiency. However, the voice response client can also access voice prompts, messages and other audio files in real-time using Internet streaming protocols such as Real-time Transfer Protocol (RTP) across the call center virtual private network.

[0041] The POP call manager creates an instance of the voice response client, and passes an incoming call to it to handle, by specifying the operation parameter ANSWER.

This results in the voice response client contacting the voice response server at the

premises call center gateway by calling AccessScript, with the parameter ANSWER_SCRIPT, to receive the script to execute. This script is passed to ExecuteScript, which processes and executes the script which typically interacts with the user presenting announcements and menu options and accepts user input via touch tones keys on the phone or speech recognition. Thereafter, when an external event occurs, the voice response client processes the event and takes actions. When user input is received through the message USER_INPUT, the input is decoded by calling ProcessUserInput, which determines what action should be performed in response to the input. This is codified by the returned result, UserRequest, which is passed to the voice response server to process by calling AccessScript. AccessScript returns back to the voice response client a new script to process, which is once again passed to ExecuteScript. If the user input is a request for operator assistance, the script returned by the business call center application running on the premises voice response server instructs the POP voice response client to transfer the call back to the local POP call manager to be queued awaiting availability of a call center agent. The POP voice response client sends the message, QUEUE_CALL, to the POP call manager. The POP call manager, as described earlier, queues the call and requests the corresponding premises call manager to generate a proxy call at the business call center ACD on its behalf. The POP call manager also creates an instance of the voice response client with the operation parameter HOLD. The general operation of the voice response client in this case is identical to the case of ANSWER, except that the first script, which is requested by calling AccessScript, is with the parameter HOLD_SCRIPT. This parameter serves to distinguish the script executed by the voice response client when the user is on hold from when the user is going through self-service.

[0042] When the control events USER_HANGUP, which is generated when a user hangs up the phone, or TERMINATE, which is sent by the POP call manager to terminate user interaction, occur, the voice response client responds by sending the messages USER_TERMINATION and TRANSFER_CALL to the POP call manager.

[0043] While the user is awaiting an available agent, the voice response client interacts with the voice response server to acquire status updates regarding the progress of proxy calls that the premises call manager had originated. Based on this information and call center configuration options, the voice response client alerts the waiting caller with status update messages. The script selected by HOLD_SCRIPT controls this behavior.

[0044] A POP network manager receives requests from the local POP call manager to translate the called party number of an incoming call to the network address of the premises call center gateway by a call to its procedure TranslateNumberToAddress. The POP network manager in turn requests the global POP network directory service to retrieve the entry corresponding to the specified called party number. The POP network manager accesses the network address of the corresponding premises call center gateway from the retrieved entry and returns it to the requesting local POP call manager. It should be noted that the global POP network directory service could be implemented using a single directory server or a collection of directory servers with replicated data for additional reliability. It should also be noted that the directory servers could be co-located at the POP call centers. As noted earlier, the directory service can be implemented using Internet standard LDAP compliant directory services.

Premises Call Manager

[0045] Referring to pseudo code listing C, appended hereto, a premises call manager receives requests from POP call managers to allocate proxy call resources at the business

call center ACD on their behalf, originate a proxy call on the allocated resources and finally bridge the incoming call from the POP call manager with the ACD line. The premises call center gateway is equipped with an inbound and an outbound voice port pair. The inbound voice port receives a call from the POP call manager and the outbound port is connected to the ACD, and looks like an incoming voice line to the ACD. When the agent is ready to take the user's call, the inbound and outbound ports are bridged to pass the bi-directional conversation. It should be noted that as far as the call center ACD is concerned, such a call appears no different than if it were to be received on one of its inbound trunks.

[0046] A POP call manager requests a premises call manager to allocate a port line pair through the message `ALLOCATE_PROXY_CALL`. Upon receiving such a request, the premises call manager locally creates a proxy call record and allocates a voice port pair for the call by calling `AllocateLinePair`. If this allocation is successful, it creates a new proxy call associated with the voice port pair and the incoming call, and returns a handle to the proxy call to the invoking POP call manager. If it is unable to allocate a line pair, it returns failure, causing the invoking POP call manager to produce a busy tone to the caller.

[0047] When the POP call manager requests the premises call manager to place the proxy call to the ACD through the message `PLACE_PROXY_CALL`, the premises call manager places the call to the ACD, noting the ACD's call ID. It returns success to the POP call manager, setting the state of the call. The premises call manager then monitors the progress of the proxy call using the ACD's CTI interface. When the proxy call is about to be delivered to a live call center agent, the call center ACD alerts the premises call manager through the message `AGENT_READY`. Upon receiving the notification,

the premises call manager identifies the POP call manager originally responsible for the proxy call and the matching inbound voice port using the proxy call record. The premises call manager then notifies the responsible POP call manager specifying a direct inward dialing (DID) number corresponding to the inbound voice port. Upon receiving such notification from the premises call manager, the receiving POP call manager identifies the matching queued call and dials the provided DID number to transfer it to the notifying premises call center gateway. When this call is received by the premises gateway, it appears to the premises call manager through the message INCOMING_CALL. Using the inbound voice port on which the call arrives at the premises call center gateway, the premises call manager matches it to the local proxy call record and bridges the call to the corresponding outbound voice port, by calling BridgeCall. This series of operations results in the customer call queued at the POP call center to be connected just in time to the appropriate business call center agent, as the agent becomes available. At any time during the call, if the premises call manager receives the message AGENT_TERMINATION from the ACD, it notifies the POP call manager of the event by sending the message AGENT_TERMINATION. The premises call manager may also receive a notification from the POP call manager that the user has terminated the call, through the message TERMINATE_CALL. In either case, it performs clean up of the voice port pairs and the proxy call record.

[0048] The above described techniques can be extended to accommodate multiple premises call center gateways at a single location. This is accomplished through one of the premises call center gateways acting as a master, selecting an appropriate gateway for handling an incoming call and returning that gateway address as part of processing INCOMING_CALL. Similarly, multiple POP-enabled business call center locations,

each with one or more premises call center gateways can also be accommodated by one of the locations acting as the master site receiving the incoming call requests.

Voice Response Server

[0049] Referring to pseudo code listing D, appended hereto, a premises voice response server hosts interactive voice response applications including voice prompts, menus, scripts and forms customized to the local business call center. The premises voice response server connects to business call center databases 198 to access customer and business information as needed by the hosted interactive voice response applications. In one preferred embodiment, the premises voice response server dynamically down loads all or parts of the requested interactive voice response applications to the requesting POP voice response client. The voice response client requests two types of scripts, one designated by the message ANSWER_SCRIPT and the other by HOLD_SCRIPT. These scripts may perform different functions for the business call center depending on whether the phone is being answered or the call is being placed in a queue awaiting an agent. Thereafter, the voice response client requests through a generic message VRC_REQUEST, which provides sufficient context such as requesting voice response client, last script executed, and new user input. In an alternative preferred embodiment, the voice response server responds to specialized request protocols such as HTTP from remote voice user interface managers embedded in POP voice response clients distributed at POP call centers. In response to such requests, the premises voice response server supplies the requested voice prompts, menus, forms and scripts to the requesting voice user interface manager. This configuration allows the voice user interface manager embedded in the POP voice response client to adapt dynamically to the interactive voice

response application customized to the business call center to which the inbound call is directed.

[0050] A premises network manager connects to the call center network for the corresponding business call center. The premises network manager initially registers with the global network directory service and creates a business call center service record for each POP call center connected to its call center network. The service record contains at a minimum the called party number at the POP call center corresponding to its business call center and the matching premises call center gateway network address. As described earlier, the POP call manager accesses this service record to identify the business call center gateway corresponding to an inbound call arriving on a particular called party number.

[0051] In a preferred embodiment, the POP call center gateway and the premises call center gateway each use two voice ports to bridge the call between the user and the long distance network, and the long distance network and the ACD. If the POP call center connects to a CTI-enabled switch, the requirement to bridge the call is eliminated in the POP call center gateway, since the POP call manager can request the switch perform a "transfer connect" through the CTI interface. Likewise, if the premises call center utilized a CTI-enabled PBX, the need to bridge the voice call at the premises is similarly eliminated.

SYSTEM MANAGEMENT

Configuration of Call Center Networks

[0052] When a business call center network is created, the POP call center network directory service entries are required to be created, as it is the configuration source to the network of all POP call centers. The business selects the participating POPs and this

determines the allocation of telephone numbers local to the POP's LATA. The set of all the telephone numbers, along with area code is also registered with the SMS/800 system to enable the point of call routing of the toll-free number. The address of the premises call center gateway server is associated with each of the telephone numbers allocated for the business. This ensures that when the call arrives, the query for the address of the premises call center gateway server is correctly handled. The query is based on the local number allocated at the POP to which the SMS/800 system referred the toll-free call. Also to be registered at the directory service is the starting point Uniform Resource Locator (URL) for any HTTP based communication, such as the POP voice response client.

Management of Audio Media in the POP

[0053] The voice response client at the POP call center gateway executes voice response application scripts that are created by the business. It is expected to handle a large number of audio media files in the course of executing these voice response application scripts. When a business changes its media files the voice response client would download large amounts of data in order to replace the cached audio resources. This process should be performed when not handling a user call. Accordingly, an auxiliary media management process in the POP call center system coordinates the validation and replacement of cached files as a maintenance task within the system.

[0054] In the foregoing specification and in the following pseudo code listings which form part of the specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made to the specific exemplary embodiments without departing from the broader spirit and scope of the invention as set forth in the appended claims.

[illegible]

Pseudo Code Listing A - POP Call Manager

[0055] POPCallManager (...) {

Messages:

[INCOMING_CALL, QUEUE_CALL, TRANSFER_CALL, AGENT_READY,
USER_TERMINATION, AGENT_TERMINATION];

Data:

```
ICR {  
  POPGatewayAddress;      // Address of the POP CC Gateway  
  InLineNumber;           // Unique Line number of the line within local system  
  CalledPartyNumber;      // Number that was called  
  CallerNumber;           // Caller's number  
  PremisesGateway;        // Structure of Addresses of the Premises Servers hosting  
                           // Call Manager Server & Voice response server and possibly others  
  ProxyCallID;            // Id of the Proxy Call registered with Call Manager Server  
  VRCInstance;            // Handle to the associated instance of Voice Response Client  
  PremisesNumber;         // Number provided by Call Manager Server to transfer call  
  OutLineNumber;          // Unique Line number of a line to use to transfer call  
}
```

Program:

```
Initialize( ... );  
While ( FOREVER ) {  
  Message = ReceiveMessage (...);  
  Switch ( Message.Operation ) {  
    Case INCOMING_CALL:      // Request to process incoming call from the PSTN  
      // Incoming Call on "LineNumber", to "CalledPartyNumber" from "CallerNumber"  
      ICR = CreateIncomingCallRecord ( ... );  
      // Set ICR LineNumber, CalledPartyNumber & CallerNumber to Call Values  
      ICR.InLineNumber = Message.LineNumber;  
      ICR.CalledPartyNumber = Message.CalledPartyNumber;  
      ICR.CallerNumber = Message.CallerNumber;  
      // Translate Called Party Number to authoritative Premises Gateway Address  
      PremisesGateway = TranslateNumberToAddress ( ICR.CalledPartyNumber, ... );  
      // Send Proxy Call Request to Premises Call Manager at PremisesGateway,  
      // passing Called Party Number and Caller Number. PremisesGateway may return  
      // the address of a different (available) gateway.  
      ProxyCall = SendMessage ( PremisesGateway.PremisesCallManager,  
                                ALLOCATE_PROXY_CALL, ... );  
      // If Proxy Call request is successful, accept the call being presented on Line Number  
      // Record the returned ID & Gateway address for subsequent communication.  
      If ( ProxyCall.Reply = SUCCESS ) {  
        ICR.ProxyCallID = ProxyCall.ID;  
        ICR.PremisesGateway = ProxyCall.PremisesGateway;  
        // Create a new Voice Response Client instance and instruct it to answer call  
        ICR.VRCInstance = CreateVoiceResponseClientInstance ( ICR, ANSWER, ... );  
        ...  
      } Else {  
        // Respond with Number Busy on Call being Presented on LineID  
        ...  
      }  
      Break;  
  
    Case QUEUE_CALL:         // Request to queue a call at Premises ACD  
      // Call is identified by Incoming Call Record, ICR  
      ICR = message.ICR;  
      // Place this call on virtual global queue with Premises Call Manager at Premises Gateway,  
      // passing ProxyCallID from ICR  
      ProxyCall = SendMessage ( ICR.PremisesGateway.PremisesCallManager,
```

```

PLACE_PROXY_CALL, ICR.ProxyCallID, ... );
// ProxyCallID now has a queue position at Premises ACD
// Create a new Voice Response Client instance and instruct it to hold the call
ICR.VRCInstance = CreateVoiceResponseClientInstance ( ICR, HOLD, ... );
...
Break;

Case AGENT_READY:      // Notification that agent is ready to receive call
// Call is identified by Proxy call identifier
ICR = FindICRFromProxyCallID ( Message.ProxyCallID, ... );
// Premises Call Manager sends the appropriate Premises Number to call
ICR.PremisesNumber = Message.PremisesNumber;
// Request Voice Response Client Instance to relinquish call
SendMessage ( ICR.VRCInstance, TERMINATE, .... );
...
Break;

Case TRANSFER_CALL:    // Request to transfer a call to Premises Call Manager
// Call is identified by ICR
ICR = Message.ICR;
// Allocate an available outbound line
ICR.OutLineNumber = AllocateLine ( );
// Place a call on allocated line to Premises Call Manager using Premises Number
PlaceCall (ICR.OutLineNumber, ICR.PremisesNumber, ... );
// When the call is accepted bridge the incoming call to outbound call
BridgeCall (ICR.InLineNumber, ICR.OutLineNumber, ... );
...
Break;

Case USER_TERMINATION: // Notification that a user has terminated a call
// Call is identified by ICR
ICR = Message.ICR;
// Inform Premises Call Manager of user termination event
SendMessage (ICR.PremisesGateway.PremisesCallManager, TERMINATE_CALL,
              ICR.ProxyCallID, ... );
// Terminate call and clean up
TerminateCall ( ICR.InLineNumber, ICR.OutLineNumber, ... );
DeleteIncomingCallRecord ( ICR );
CleanUp ( );
...
Break;

Case AGENT_TERMINATION: // Notification that an agent has terminated a call
// Call is identified Proxy call identifier
ICR = FindICRFromProxyCallID ( Message.ProxyCallID, ... );
// Terminate the inbound and outbound calls and clean up
TerminateCall ( ICR.InLineNumber, ICR.OutLineNumber, ... );
DeleteIncomingCallRecord ( ICR );
CleanUp ( );
...
Break;
}
} // End POPCallManager

```

Pseudo Code Listing B - Voice Response Client

```
[0056] VRCInstance ( ICR, Operation, ... ) {
Messages:
    [USER_INPUT , USER_HANGUP, TERMINATE];
Data:
    ICR { ... };
    Operation: [ANSWER, HOLD];
Program:
    Initialize ( ... );
    // Access the Voice Response Server at Premises Gateway for Starting Script
    VoiceResponseServer = ICR.PremisesGateway.VoiceResponseServer;
    If (Operation = ANSWER) {
        // If invoked to answer the call, access the answering script from Voice Response Server
        NewScript = AccessScript (VoiceResponseServer, ANSWER_SCRIPT, ... );
    } Else {
        // If invoked to hold the call, access the hold script from Voice Response Server
        NewScript = AccessScript (VoiceResponseServer, HOLD_SCRIPT, ... );
    }
    // Execute the accessed script on appropriate line
    Status = ExecuteScript ( ICR.LineNumber, NewScript, ... );
    // If the call needs to be queued, inform the POP Call Manager
    If ( Status = QUEUE_CALL ) {
        SendMessage ( POPCallManager, QUEUE_CALL, ... );
        CleanUp ( );
        Exit ( ... );
    }
    While ( FOREVER ) {
        Message = Receive.Message ( ... );
        Switch ( Message.Operation ) {
            Case USER_INPUT:                // User input in the form of touch tones or speech recognition
                // Process user input and access additional scripts from Voice Response Server, if needed
                UserRequest = ProcessUserInput (Message, ... );
                NewScript = AccessScript (VoiceResponseServer, UserRequest, ... );
                // Execute the accessed script on appropriate line
                Status = ExecuteScript ( ICR.LineNumber, NewScript, ... );
                // If the call needs to be queued, inform the POP Call Manager
                If ( Status = QUEUE_CALL ) {
                    SendMessage ( POPCallManager, QUEUE_CALL, ICR, ... );
                    CleanUp ( );
                    Exit ( ... );
                }
                ...
                Break;

            Case USER_HANGUP:                // User hangs up the call
                // Inform the POP Call Manager that the user has terminated the call
                SendMessage ( POPCallManager, USER_TERMINATION, ICR, ... );
                // Clean up and exit
                CleanUp ( ... );
                Exit ( ... );

            Case TERMINATE:                // POP Call Manager terminates hold session
                // Transfer call back to POP Call Manager
                SendMessage ( POPCallManager, TRANSFER_CALL, ICR, ... );
                // Clean up and exit
                CleanUp ( ... );
                Exit ( ... );
        }
    }
}
```

```
    }  
  }  
} // End VRCInstance
```

2025-04-24 10:00:00

Pseudo Code Listing C - Premises Call Manager

[0057] PremisesCallManager (...) {

Messages:

[ALLOCATE_PROXY_CALL, PLACE_PROXY_CALL, AGENT_READY, INCOMING_CALL,
USER_TERMINATION, AGENT_TERMINATION];

Data:

```
PCR {  
    POPGateway;           // Address of the POP Gateway responsible for this call  
    CalledPartyNumber;     // Number that was originally called by the user  
    CallerNumber;          // Caller's number  
    // Total number of Line pairs limit the maximum number of active calls at all POPs  
    LinePair {  
        InLineNumber;     // Line number on which POP Gateway call arrives  
        OutLineNumber;    // Line number on which ACD call is placed  
    };                    // Lines are allocated in pairs.  
    ProxyCallStatus;       // Status of the proxy call in Premises Call Manager  
    ProxyCallID,           // ID of the proxy call that is provided to the POP call manager  
    ACDCallStatus;         // Status of the proxy call in the ACD  
    ACDInLineNumber;      // ACD Line number on which outbound call is placed  
    ACDCallID;             // ACD Call handle  
} // Proxy Call Record
```

Program:

```
Initialize ( ... );  
While ( FOREVER ) {  
    Message = ReceiveMessage ( ... );  
    Switch ( Message.Operation ) {  
    Case ALLOCATE_PROXY_CALL:      // Request from a POP for a proxy call allocation  
        // Allocate a line pair. There must be as many line pairs as there are inbound ACD ports  
        // If successful, then create proxy call record and assign the line pair to it. AllocateLinePair  
        // could be extended to return the address of an available premises call center  
        // gateway if this gateway does not have available line pairs.  
        Status = AllocateLinePair ( LinePair, ... );  
        If ( Status = SUCCESS ) {  
            // Proxy call identifier is assigned to the proxy call record  
            PCR = CreateProxyCallRecord ( ... );  
            // Assign allocated line pair to proxy call record  
            PCR.LinePair = LinePair;  
            // Assign POP Gateway address, Called Party Number and Caller Number to Proxy  
            // Call record  
            PCR.POPGateway = Message.POPGateway;  
            PCR.CalledPartyNumber = Message.CalledPartyNumber;  
            PCR.CallerNumber = Message.CallerNumber;  
            PCR.ProxyCallID = &PCR;    // address handle to PCR  
            // Reply to POP Call Manager indicating success and pass Proxy call identifier  
            SendMessage ( PCR.POPGateway.POPCallManager, SUCCESS,  
                          PCR.ProxyCallID, ... );  
            PCR.ProxyCallStatus = ALLOCATED;  
        } Else { // Reply to POP Call Manager indicating rejection of incoming call  
            SendMessage ( Message.POPGateway.POPCallManager, FAILURE, ... );  
        }  
        ...  
        Break;  
  
    Case PLACE_PROXY_CALL: // Place a proxy call to the ACD on behalf of requesting POP  
        // Call is identified by PCR which is identified by the Proxy call identifier  
        PCR = FindPCRFromProxyCallID ( Message.ProxyCallID, ... );  
        // Place a proxy call to ACD on Out Line Number assigned to Proxy Call Record
```

```

ProxyCall = PlaceProxyCall ( PCR.LinePair.OutLineNumber, ... );
// Set Proxy Call Record ACD call identifier and Line Number
PCR.ACDCallID = ProxyCall.ACDCallID;
PCR.ACDInLineNumber = ProxyCall.ACDInLineNumber;
// Reply to POP Call Manager indicating successful queuing of call at ACD
SendMessage ( PCR.POPGateway.POPCallManager, SUCCESS, PCR.ProxyCallID, ... );
PCR.ProxyCallStatus = QUEUED;
...
Break;

```

```

Case AGENT_READY:          // Notification from ACD that the agent is ready to take call
// Call is identified by PCR which is identified by ACD inbound line number
PCR = FindPCRFromACDInLineNumber ( Message.ACDInLineNumber, ... );
// Translate inbound line number to phone number to be called for transfer of call
PremisesNumber = TranslateLineToPhoneNumber ( PCR.LinePair.InLineNumber );
// Signal POP Call Manager that agent is ready to accept call on DID, PremisesNumber
SendMessage ( PCR.POPGateway.POPCallManager, AGENT_READY,
              PCR.ProxyCallID, PremisesNumber );
PCR.ProxyCallStatus = AGENT_READY;
...
Break;

```

```

Case INCOMING_CALL:        // Request to process an incoming call from POP Gateway
// Call is identified by PCR which is identified by inbound line number
// Call can only be from the authorized POP Gateway
PCR = FindPCRFromInLineNumber ( Message.InLineNumber, ... );
// Bridge the call from POP Gateway to ACD proxy call
BridgeCall (PCR.LindPair.InLineNumber, PCR.LinePair.OutLineNumber, ... );
// Reply to POP Call Manager indicating successful bridging of call to ACD
SendMessage ( PCR.POPGateway.POPCallManager, SUCCESS, PCR.ProxyCallID, .. );
PCR.ProxyCallStatus = ACTIVE;
...
Break;

```

```

Case USER_TERMINATION:    // Notification that a user terminated a call
// Call is identified by PCR which is identified by Proxy call identifier
PCR = FindPCRFromProxyCallID ( Message.ProxyCallID, ... );
// Terminate the call at ACD and clean up
TerminateCall (PCR.ACDInLineNumber, ... );
DeleteProxyCallRecord ( PCR );
CleanUp ( );
...
Break;

```

```

Case AGENT_TERMINATION:   // Notification that an agent terminated a call
// Call is identified by PCR which is identified by ACD inbound line number
PCR = FindPCRFromACDInLineNumber ( Message.ACDInLineNumber, ... );
// Signal POP Call Manager that agent has terminated call
SendMessage ( PCR.POPGateway.POPCallManager, AGENT_TERMINATION,
              PCR.ProxyCallID, ... );
// Terminate the call at ACD and clean up
TerminateCall (PCR.ACDInLineNumber , ... );
DeleteProxyCallRecord ( PCR );
CleanUp ( );
...
Break;

```

```

}

```

```

}

```

Table 1. (continued)	
1.0	4.0
0.5	2.0
0.25	1.0
0.125	0.5
0.0625	0.25
0.03125	0.125
0.015625	0.0625
0.0078125	0.03125
0.00390625	0.015625
0.001953125	0.0078125
0.0009765625	0.00390625
0.00048828125	0.001953125
0.000244140625	0.0009765625
0.0001220703125	0.00048828125
0.00006103515625	0.000244140625
0.000030517578125	0.0001220703125
0.0000152587890625	0.00006103515625
0.00000762939453125	0.000030517578125
0.000003814697265625	0.0000152587890625
0.0000019073486328125	0.00000762939453125
0.00000095367431640625	0.000003814697265625
0.000000476837158203125	0.0000019073486328125
0.0000002384185791015625	0.00000095367431640625
0.00000011920928955078125	0.000000476837158203125
0.000000059604644775390625	0.0000002384185791015625
0.0000000298023223876953125	0.00000011920928955078125
0.00000001490116119384765625	0.000000059604644775390625
0.000000007450580596923828125	0.0000000298023223876953125
0.0000000037252902984619140625	0.00000001490116119384765625
0.00000000186264514923095703125	0.000000007450580596923828125
0.000000000931322574615478515625	0.0000000037252902984619140625
0.0000000004656612873077392578125	0.00000000186264514923095703125
0.00000000023283064365386962890625	0.000000000931322574615478515625
0.000000000116415321826934814453125	0.0000000004656612873077392578125
0.0000000000582076609134674072265625	0.00000000023283064365386962890625
0.00000000002910383045673370361328125	0.000000000116415321826934814453125
0.000000000014551915228366851806640625	0.0000000000582076609134674072265625
0.0000000000072759576141834259033203125	0.00000000002910383045673370361328125
0.00000000000363797880709171295166015625	0.000000000014551915228366851806640625
0.000000000001818989403545856475830078125	0.0000000000072759576141834259033203125
0.0000000000009094947017729282379150390625	0.00000000000363797880709171295166015625
0.00000000000045474735088646411895751953125	0.000000000001818989403545856475830078125
0.000000000000227373675443232059478759765625	0.0000000000009094947017729282379150390625
0.0000000000001136868377216160297393798828125	0.00000000000045474735088646411895751953125
0.00000000000005684341886080801486968994140625	0.000000000000227373675443232059478759765625
0.000000000000028421709430404007434844970703125	0.0000000000001136868377216160297393798828125
0.0000000000000142108547152020037174224883515625	0.00000000000005684341886080801486968994140625
0.00000000000000710542735760100185871124417578125	0.000000000000028421709430404007434844970703125
0.000000000000003552713678800500929355622087890625	0.0000000000000142108547152020037174224883515625
0.0000000000000017763568394002500461778110439453125	0.00000000000000710542735760100185871124417578125
0.00000000000000088817841970012502308890552197265625	0.000000000000003552713678800500929355622087890625
0.000000000000000444089209850062511544452760986328125	0.0000000000000017763568394002500461778110439453125
0.0000000000000002220446049250312557722263804931640625	0.00000000000000088817841970012502308890552197265625
0.00000000000000011102230246251562788611319024658203125	0.000000000000000444089209850062511544452760986328125
0.000000000000000055511151231257813943056595123291	

Pseudo Code Listing D - Voice Response Server

```
[0058] VoiceResponseServer ( ... ) {
Messages:
    [ANSWER_SCRIPT, HOLD_SCRIPT, VRC_REQUEST];
Data:
Program:
    Initialize ( ... );
    While ( FOREVER ) {
        Message = ReceiveMessage ( ... );
        Switch ( Message.Operation ) {
            Case ANSWER_SCRIPT: // Request for starting script for answering a call
                // Voice Response Client instance is identified by message
                VRCInstance = Message.VRCInstance;
                // Access the starting script for answering a call and send it to VRC instance
                NewScript = AccessScript ( ANSWER_SCRIPT, ... );
                SendMessage ( VRCInstance, NewScript, ... );
                ...
                Break;

            Case HOLD_SCRIPT: // Request for starting script for servicing a call during hold
                // Voice Response Client instance is identified by message
                VRCInstance = Message.VRCInstance;
                // Access the starting script for servicing a call on hold and send it to VRC instance
                NewScript = AccessScript ( HOLD_SCRIPT, ... );
                SendMessage ( VRCInstance, NewScript, ... );
                ...
                Break;

            Case VRC_REQUEST; // Request for a script based on current user interaction
                // Voice Response Client instance is identified by message
                VRCInstance = Message.VRCInstance;
                // Access the script for servicing the user request and send it to VRC instance
                NewScript = AccessScript ( Message.UserRequest, ... );
                SendMessage ( VRCInstance, NewScript, ... );
                ...
                Break;
        }
    }
} // End VoiceResponseServer
```